



US Army Corps  
of Engineers®

## A Custom Data Logger for Real-Time Remote Field Data Collections

*by Patrick J. Dickhudt and Kent K. Hathaway*

**PURPOSE:** The purpose of this Coastal and Hydraulics Engineering Technical Note (CHETN) is to describe a versatile, reliable, and relatively inexpensive custom-built datalogger system utilized by the Coastal and Hydraulics Laboratory (CHL), Field Research Facility (FRF), for remote real-time data collections. This custom data logger is compact and energy efficient but has the same functionality as a personal computer allowing flexibility in programming languages and data collection routines, easy incorporation of new instrumentation, simultaneous collection of numerous (20+) instruments, secure communication and data transfer, and direct communication with deployed instruments.

**INTRODUCTION:** Real-time data collections offer many advantages:

1. Instrument failures can be rapidly detected and repaired, thereby minimizing data gaps.
2. Data are passed from the instrument to a data logger, then to a land-based station minimizing chances of data loss compared to instruments that log internally and may be damaged or lost.
3. Data can be evaluated during a deployment, and settings can be adjusted to optimize resolution of processes of interest.
4. Other agencies or the public may be interested and may benefit from the available data (e.g., wind or wave data) for emergency services, navigation, education, and recreation.

The U.S. Army Corps of Engineers (USACE), CHL, FRF, had a need for a remote real-time data collection system to control instruments and log and communicate data from five observing stations in the Currituck Sound Estuary, NC<sup>1</sup>. These stations, referred to as the Currituck Sound Array (CSA), collect a suite of meteorological and oceanographic data including wind, air temperature, humidity, incoming solar radiation (above and below water), waves, currents, water level, salinity, and water temperature, as well as turbidity and many other water quality parameters. This array of instruments has a variety of control commands, sample routines, and output data formats. Additionally, the CSA was designed to act as a natural laboratory for estuarine research and as an instrument and model test bed. These capabilities required a reliable and flexible system that would allow easy modification of sampling schemes, the ability to log as many as 15 instruments with a single logger, and allow the incorporation of additional and novel instrumentation with minimal effort and expense.

---

<sup>1</sup> Wadman, H. M., and P. J. Dickhudt. In preparation. Currituck sound monitoring array. ERDC-CHL Technical Report. Vicksburg, MS: U.S. Army Engineer Research and Development Center.

Commercially available off-the-shelf loggers were tested, but they were found to be inadequate as follows:

1. They did not provide secure communication and data transmission to a shore station. They typically relied on outdated protocols such as telnet that did not provide a secure connection to communicate and transfer data.
2. They did not have enough serial communication ports to control and log the required instrumentation. As a result, multiple loggers would need to be “daisy chained,” or expensive peripherals would need to be added.
3. They did not allow direct remote communications to instruments. While they technically have this capability, when tested over a cellular modem, there was too much latency for this feature to function properly or effectively. Direct communication to instruments is desirable because it allows remote verification of instrument functionality, trouble shooting, and programming modifications.

**Description.** As a result of these shortcomings of off-the-shelf loggers, custom-built loggers were designed and fabricated. These custom loggers were built upon single board computers (SBC) running the Linux operating system. They effectively have the same functionality as a personal computer, overcoming many of the limitations of off-the-shelf loggers. Additionally, off-the-shelf loggers typically operate on a very limited set of commands. These custom Linux-based loggers have a much more diverse and powerful selection of commands overcoming many of the unique challenges of real-time data collection with robust code and programmatic “watchdogs” that can automatically make sure the logger, instruments, and communications are operating as intended.

**Hardware.** The “brains” of the dataloggers are Technologic Systems TS-7250-V2 SBCs. While there are many suitable SBCs on the market, these particular units were chosen based on size, cost, processing power, number of available analog and serial ports, ruggedness, and operating temperature range.

Remote data logging units are typically deployed at exposed locations and housed in small weather-proof enclosures requiring them to be compact, energy efficient, and able to operate at temperature extremes. The TS-7250-V2 is designed for industrial applications making it rugged and reliable. They are compact (10 centimeters [cm] × 11 cm) and operate over a large temperature range (-40° C to 85° C). They are energy efficient, drawing approximately 200 milliamps (mA) at 12 volts (V).

**RS232 and Analog-to-Digital Capabilities.** Most meteorologic and oceanographic instruments output an analog data signal or communicate via serial communication protocols (e.g., RS232 or RS485). The TS-7250-V2 comes equipped with five channels of analog-to-digital conversion (ADC) and three serial ports. Additionally, up to three PC104 peripheral boards (Figure 1) can be added to expand the ADC or serial communication capabilities. Each additional board can add 12 or more ADC channels or four serial ports. In the present configuration of this data logger, three TSER4 serial communication boards (also from Technologic Systems) have been added to expand the logging capabilities to a total of five ADC channels and 14 serial communication ports. One serial port is reserved for direct communications with the unit. In this configuration, all serial communications are RS232.

However, one of the SBC serial ports and half of the peripheral add-on serial ports are capable of RS422/RS485 communication as well. If required, an additional eight RS232, RS422, or RS485 ports can be added by utilizing USB-to-serial conversion units plugged into the two USB ports available on the TS-7250-V2. Future Technologies Devices International (FTDI) USB-to-serial converters are recommended as they have been thoroughly tested and found reliable and easy to use. If plugged in before powering up the logger, they are automatically recognized as serial ports, and the port naming convention is static. With the available PC104 peripherals and USB-to-serial converters, these custom loggers are capable of logging five ADC channels and 22 RS232 serial ports while drawing approximately 300 mA of current. The SBC, three PC104 peripheral serial boards, and two USB-to-serial boards described here retailed for a total of approximately \$650 in 2016.

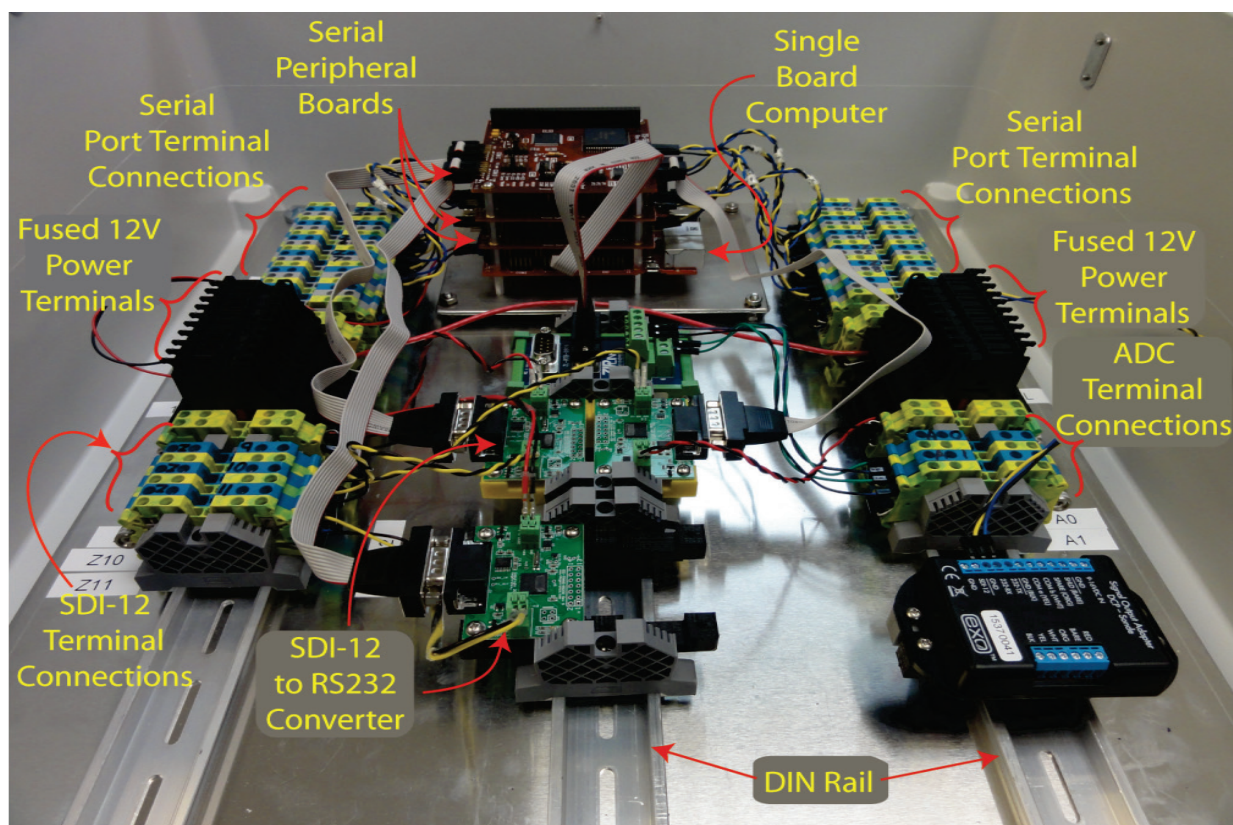


Figure 1. Labeled photograph of datalogger and components.

**SDI-12 Communications.** While standard RS232 or RS485 serial protocols are the most common, some instruments require SDI-12 (serial data interface at 1200 baud) communication. This can be accommodated by adding SDI-12 to RS232 converter units (Figure 1), which range in price from approximately \$125 to \$400 each. SDI-12 communications ostensibly have the advantage of allowing multiple (10 or more) sensors to communicate over a single serial port. However, accommodating multiple sensors with different sampling schemes and data output formats results in convoluted data collection code. In general, the advantages of communicating over a single serial port are greatly diminished by convoluted code and a loss in sample scheme flexibility. Thus, the logger shown in Figure 1 uses a serial port and SDI-12-to-RS232 converter for each instrument requiring SDI-12 communication.

**Terminal Connections and Lightning Protection.** Connections to the analog and serial ports on the SBC and serial peripheral boards are made via  $2 \times 5$  pin headers. To facilitate easy and organized connection of instruments and lightning protection components, the  $2 \times 5$  headers are connected to a series of terminal connection blocks mounted to standard metal rails known as DIN rails (Figure 1). DIN rail terminal blocks and components are available in a multitude of configurations and colors from numerous manufacturers. Two types were chosen for this application: fused terminal blocks were used for all 12 V power connections, and four-conductor terminal blocks were utilized for all signal and ground connections. The fused terminal blocks (Figures 1 and 2) provide a neat and compact means of providing power to instruments and logger components while providing overcurrent protection specific to the needs of each component. Each fused terminal contains a user-selectable and replaceable  $5 \times 20$  millimeter fuse. Jumpers can interconnect the fused terminal blocks allowing power to be easily distributed amongst numerous terminals. Other four-conductor terminal blocks offer two locations for making the signal connection and two for adding lightning protection. Every wire that connects to an instrument or other outside component is protected from lightning via a transient voltage suppression diode (transorb) and a gas discharge tube connecting the signal terminal to a neighboring ground terminal (Figure 2). The transorbs clamp transients just above the operational voltage, and the gas-discharge tubes protect against larger, more catastrophic (higher current) transients. This approach to lightning protection offers two levels of protection to each instrument and the logger and allows quick, inexpensive, and easy repair and recovery in the event of a lightning strike.

**Programming and Data Collection.** The TS-7250-V2 single board computers are based on the Marvell PXA168 1066MHz ARM CPU and run Debian Linux v2.6 kernel (or optionally v3.14). These SBCs allow programming flexibility by supporting many programming languages including C, C++, PERL, PHP, SH, Java, BASIC, TCL, and Python.

The USACE, CHL-FRF, has a long history of real-time data collections in the Atlantic Ocean. This has led to the development, testing, and refinement of robust and reliable data collection code primarily utilizing the C and PERL programming languages. The ability of the TS-7250-V2s to recognize these programming languages allowed data collections on the custom-built loggers to be built upon existing code that has been rigorously tested. This provided a huge savings in programming time and effort as well as providing reliable code from the start.

The majority of data collection code was written in the PERL language. A general template was developed from existing code and modified as required to fit the needs of specific instruments. The general format of this template is as follows:

1. Initialize and define variables such as sample intervals, instrument data output formats, data file headers, communications (COM) port settings, and calibration coefficients.
2. Define subroutines such as serial port line capture and sample timing determination.
3. Collect data within a loop that sends commands to the logger or sensor as necessary, captures data, applies instrument-specific calibrations as required, writes the data to a text file, and waits until it is time for the next sample. Additionally, within the data collection loop, the existing file is closed, and a new file opened each hour.
4. Complete the data collection cycle by closing data files and serial ports and cleanly exiting programs.



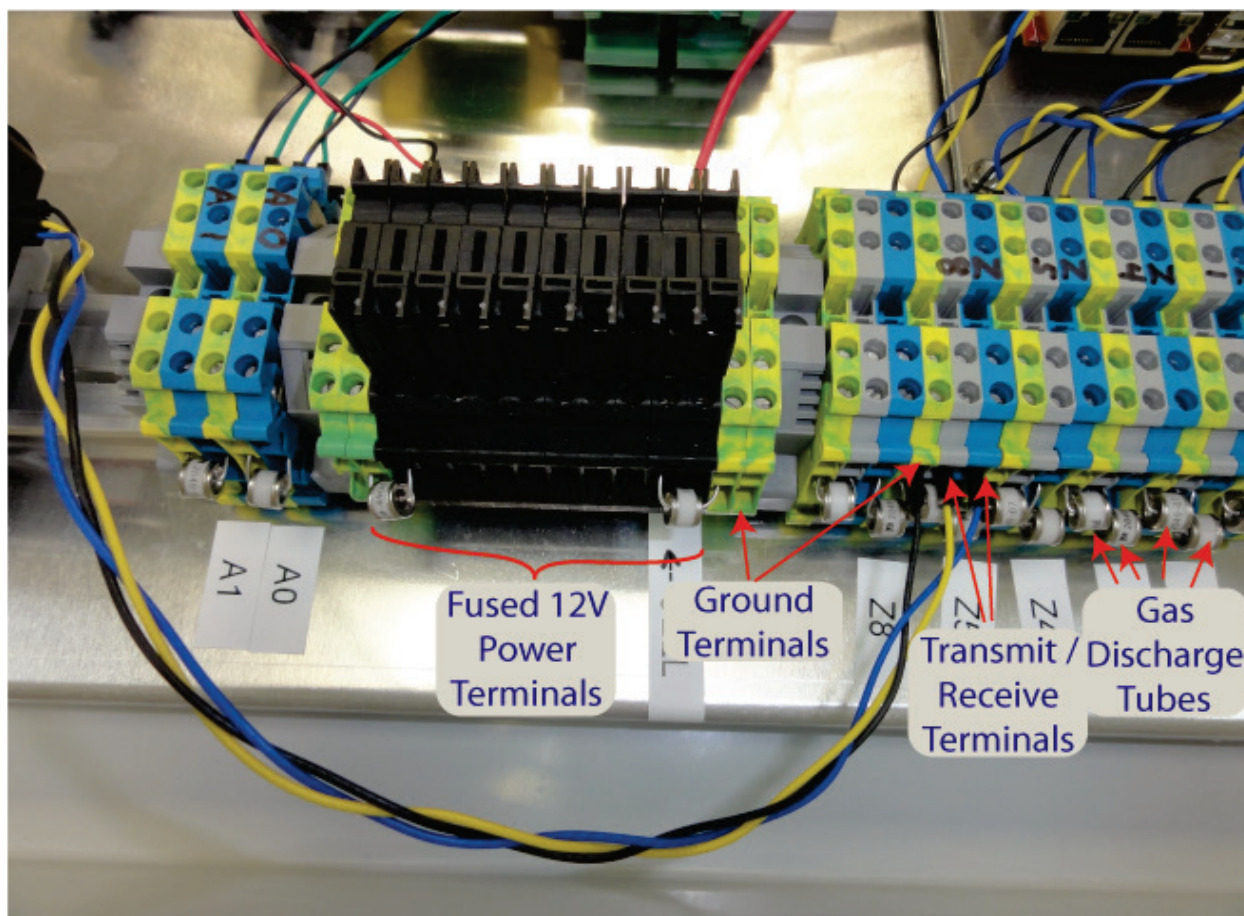


Figure 2. Close up of datalogger DIN rail components illustrating power distribution, signal and ground connection terminals, and lightning protection components. Note transient voltage suppressors are opposite gas discharge tubes in a similar configuration.

In the case of instruments that communicate their data via ADC, a simple system call will instruct the logger to display the readings from the five ADC channels. The ADC readings are captured, instrument-specific calibrations are applied, and the data are written to an hourly file.

Serial and SDI-12 instruments rely on the Perl SerialPort module to communicate with instruments via the data logger's serial ports. Instrument-specific serial port settings (baud rate, parity, data bits, handshaking) are hard coded within the program. The COM port location is specified as an input when the collection program is called. This allows flexibility in how the instruments are connected or how many instruments of a specific type are controlled by a given logger without requiring any modifications to the data collection program. For serial instruments operating in a polled mode, a command is sent over the serial port instructing the instrument to collect a sample or to start sampling if multiple samples are to be collected. The data streamed over the serial port are captured by the data logger, the data format is verified to ensure good data have been collected, calibrations are applied as appropriate, and the data are written to an hourly file. For SDI-12 instruments, a "collect data" command is sent from the serial port, through the RS232 to SDI-12 converter unit to the instrument. The logger then waits for an instrument-specific period while the instrument collects a sample and then sends a "send data command." The data then pass through the RS232 to SDI-12 converter to the data logger serial

port and are captured by the logger, the data format is verified to ensure good data have been collected, and the data are written to an hourly file.

**Communications and Data Telemetry.** Remote communications from a land-based personal computer to the data logger are achieved with a cellular router. The cellular router has a port forwarding feature that effectively makes it transparent during communications. Using secure shell network protocols (SSH), a user can securely connect through the cell router directly to the data logger. This interaction with the datalogger behaves exactly as if the user were working from the command line of a Linux PC giving a remote user complete control of the data logger, its file system, and its programs. Additionally, while operating the datalogger via SSH, a user can utilize a terminal emulator on the logger for direct connection to deployed instruments. Thus, from a remote, land-based station, a user can verify operation, troubleshoot, and reprogram deployed instruments.

An advantage of this custom data logger running the Linux operating system (as compared to commercially available dataloggers) is that files can be securely exchanged between the datalogger and remote computers using the Secure File Transfer Protocol (SFTP). SFTP allows the use of RSA encryption keys to ensure secure connections and data transfers without the need to manually enter passwords. This is not only convenient but also facilitates the reliable, secure, automated file transfers required of remote data collection systems that need to interact with government networks. Using SFTP called from automated programs running at the CHL-FRF, files are automatically transferred from the remote loggers once per hour. In addition, file management programs running on the dataloggers maintain copies of all files for a user-specified number of days as backups in the event of data transfer problems. This has proven to be a reliable and robust approach to data handling, providing months of data transfers to date with no problems.

**CONCLUSIONS:** A custom-built, data collection system was designed, assembled, programmed, and tested to facilitate real-time, remote datalogging at the FRF. While this unit is referred to as a datalogger, it is really a compact, energy-efficient, rugged computer built upon the Linux operating system. This system can be programmed using numerous programming languages, and a single unit can control and log data from more than 20 instruments simultaneously. While there was some overhead that went into developing this unit, it now represents an attractive alternative to off-the-shelf loggers and can provide flexible and reliable file handling, secure communications, and interactive or automated remote control of deployed instruments. This datalogger (Figure 1) is a powerful data collection platform that could be a valuable resource to USACE researchers and planners. A variety of data collection programs have been developed making available a library of code that can easily be adapted to a wide range of additional instruments. A single unit takes approximately 1 day to assemble and program and costs approximately \$1000 for all hardware (~\$350) and electronics (~\$650).

**ADDITIONAL INFORMATION:** For additional information, contact Patrick J. Dickhudt, Coastal Hydraulics Laboratory (CHL), Coastal Observation and Analysis Branch (COAB), 1261 Duck Road, Duck, NC 27949, at 252-261-6840 (x223) or by email: [Patrick.j.dickhudt@usace.army.mil](mailto:Patrick.j.dickhudt@usace.army.mil). The authors will share programs, detailed parts lists, and experiences with designing and building these dataloggers so that their efforts in designing, programming, and testing these dataloggers can provide a useful tool to the USACE community and other government researchers.

This CHETN should be cited as follows:

Dickhudt, P. J., and K. K. Hathaway. 2017. *A custom data logger for real-time remote field data collection*. ERDC/CHL CHETN-VI-46. Vicksburg, MS: U.S. Army Engineer Research and Development Center.  
<http://dx.doi.org/10.21079/11681/21631>.

**NOTE:** *The contents of this technical note are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such products.*